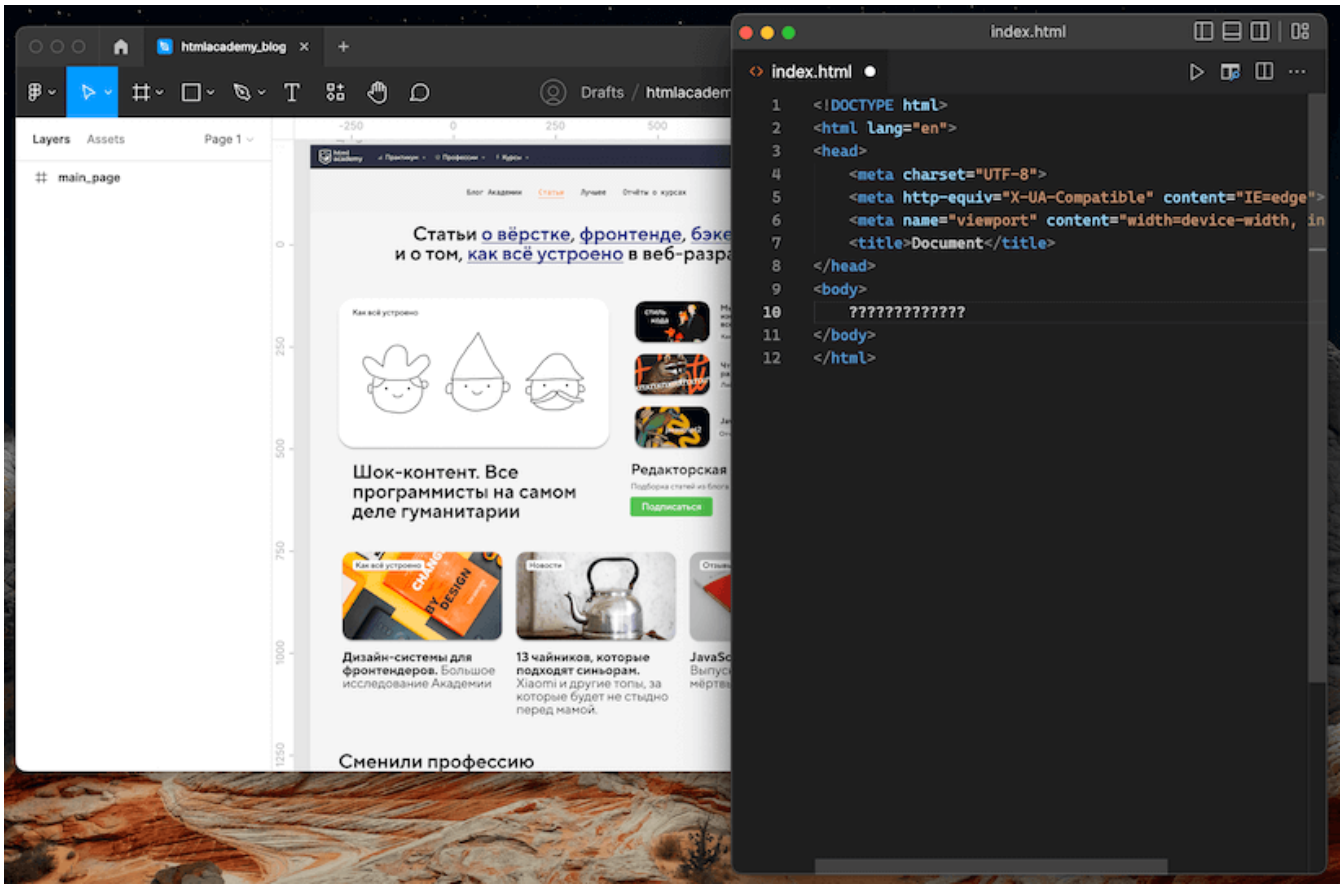
Как сверстать макет. Пошаговый план

6 октября 2022
17 211

Автор Анастасия Кузнецова

CSS

Вы открыли макет в Фигме и редактор кода. Сейчас расскажем, что нужно делать дальше, чтобы не впасть в прокрастинацию и всё сверстать.



Осмотрите макет

Зачем. Чтобы потом не отвлекаться от вёрстки.

Смотрите макет по принципу «Снаружи — внутрь» — двигаясь от крупных смысловых элементов к деталям дизайна. Чтобы было удобнее, сделайте дубликат макета в Фигме и пишите там заметки о том, что нашли.

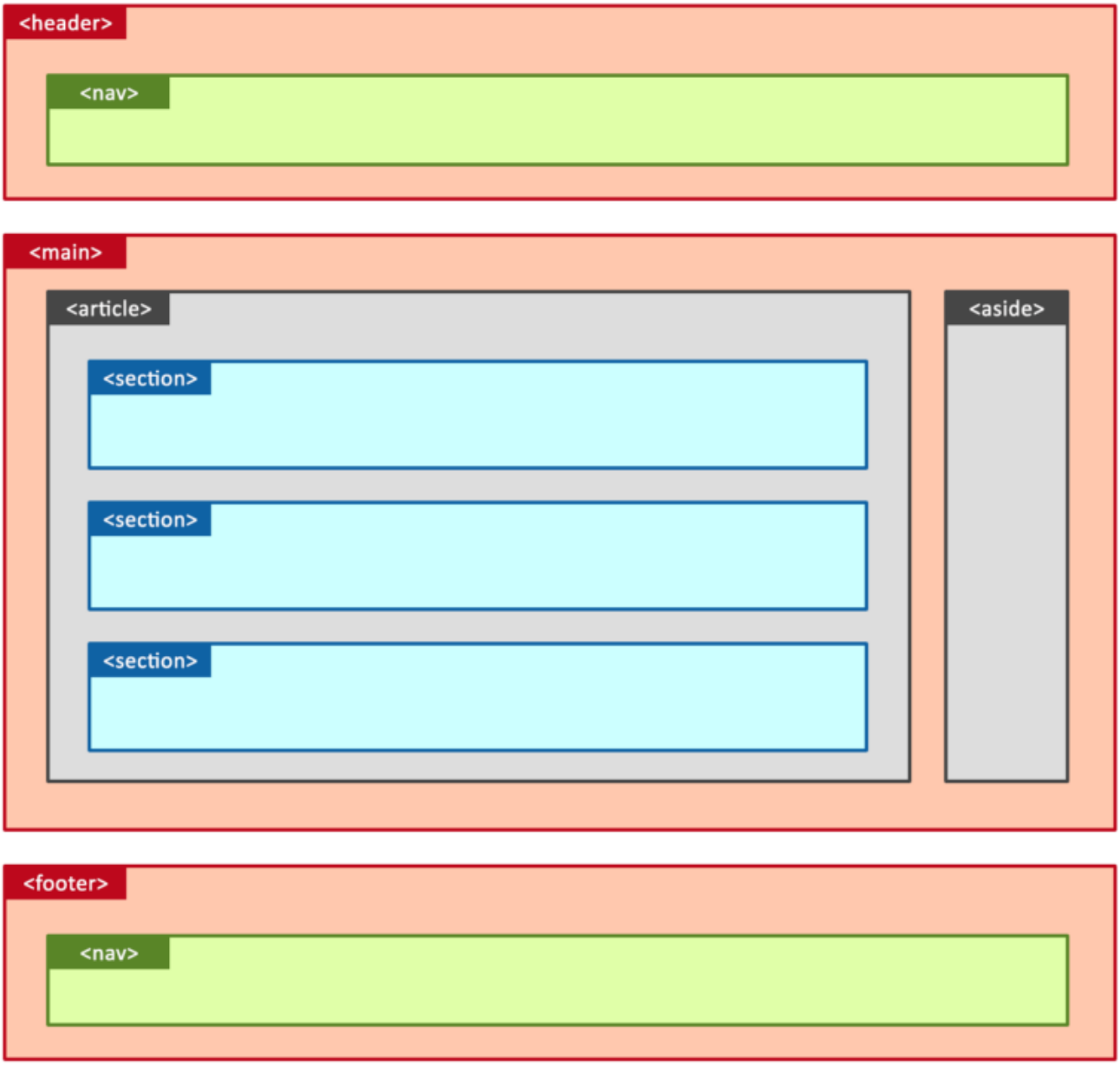
Отметьте крупные смысловые блоки и разделы. Посмотрите на страницу и выделите крупные смысловые блоки. Базовая структура любого макета состоит из трех основных тегов:

— `<header>` — шапка сайта, одинаковая на всех страницах.

— `<main>` — уникальный контент;

— `<footer>` — подвал, одинаковый на всех страницах.

Теперь ищем смысловые разделы внутри этих блоков. Поможет схема:

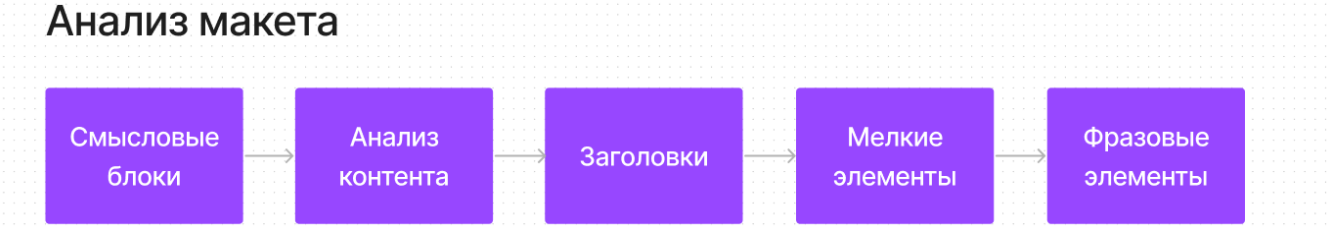


Проанализируйте контент. Присмотритесь к тексту на макете, какую функцию он выполняет? Может быть то, что вы видите, это не текст, а кнопка или раскрывающиеся меню?

На этом этапе отметьте:

1. Заголовок всего документа и заголовки смысловых разделов. Теги: `<h1>-<h6>`.
2. Мелкие элементы в смысловых разделах. Это списки, таблицы, демо-материалы, параграфы и переносы, формы, цитаты, контактная информация и прогресс.
3. Фразовые элементы. Изображения, ссылки, кнопки, видео, время и мелкие текстовые элементы.

Когда вы закончите работу, у вас в заметках будет готовая схема вёрстки и уже ничего не будет отвлекать вас от кода.

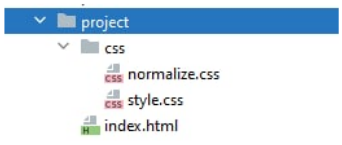


Настройте редактор кода и проект

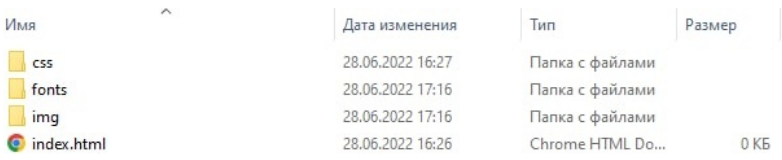
Установите редактор Visual Studio Code (или любой другой), если ещё этого не сделали, и плагин editorconfig. Он помогает разным разработчикам писать код в проекте в одном стиле.

Структура проекта. Создайте папку и положите туда файл index. html, папку css с файлами style. css и normalize. css, а также папки для картинок и шрифтов. Получится так:

Структура проекта в VS Code



Структура проекта на компьютере



Разметка

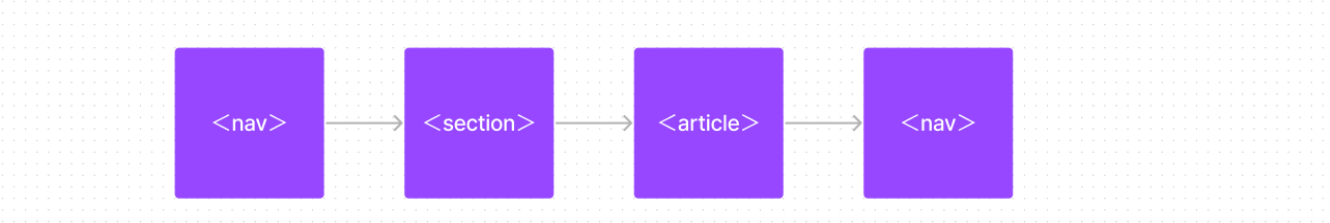
Рабочую среду подготовили, приступаем к разметке.

1. Создаем разметку страницы в файле index. html. Изображения пока не подключаем — этим займемся на этапе работы с графикой;
2. Далее, в файле при помощи нужных тегов: прописываете весь текст, расставляете все ссылки и кнопки. Нужен только HTML-код, стили пока делать не нужно.
3. Теперь определим `!DOCTYPE`, укажем язык содержимого, кодировку и заголовок страницы во вкладке браузера.

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Заголовок страницы</title>
  </head>
  <body>
  </body>
</html>
```

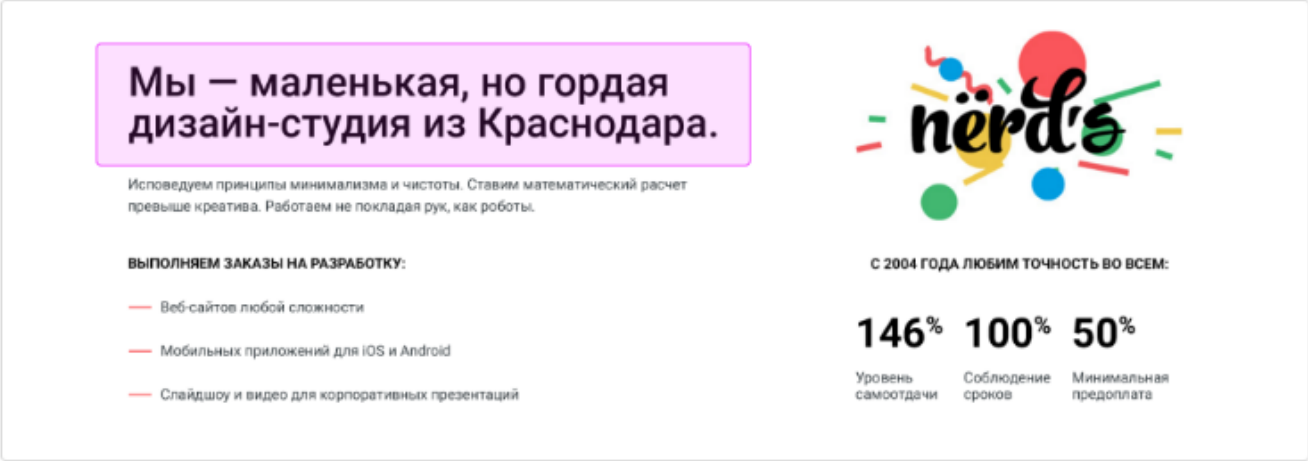
Выделяем крупные смысловые блоки на каждой странице сайта. Это `<header>` — шапка, `<footer>` — подвал и `<main>` — основное содержимое.

Размечаем в блоках крупные смысловые разделы. Выделяем главную навигацию `<nav>`, секции через `<section>`, смысловые разделы через `<article>` и дополнительное содержимое через `<nav>`.



Выделяем заголовок всего документа и заголовки смысловых разделов. Заголовок — это обманчиво простой тег. Главная проблема с заголовками такая: не всегда то, что кажется заголовком, им является.

Например, текст про дизайн-студию из Краснодара прикидывается заголовком, но на самом деле это не он:



Это само содержание, а не его резюме. Хорошим заголовком для этого блока был бы текст «О нас» или «О студии».

Размечаем мелкие элементы в смысловых разделах. Списки `<ul>` и `<ol>`, таблицы `<table>`, демонстрационные материалы `<figure>`, параграфы `<p>` и переносы `<br>`, формы `<form>`, цитаты `<blockquote>`, контактную информацию, прогресс и измерения.

Определить, какие теги использовать, можно методом исключения:

— Получилось найти самый подходящий смысловой тег — использовать его.

— Для потоковых контейнеров — `<div>`.

— Для мелких фразовых элементов (слово или фраза) — `<span>`.

Размечаем фразовые элементы. Изображения `<img>`, ссылки `<a>`, кнопки `<button>`, видео-контент `<video>`, время `<time>`, мелкие текстовые элементы `<span>`, `<b>` или `<i>`.

Разметка в редакторе кода выглядит так:

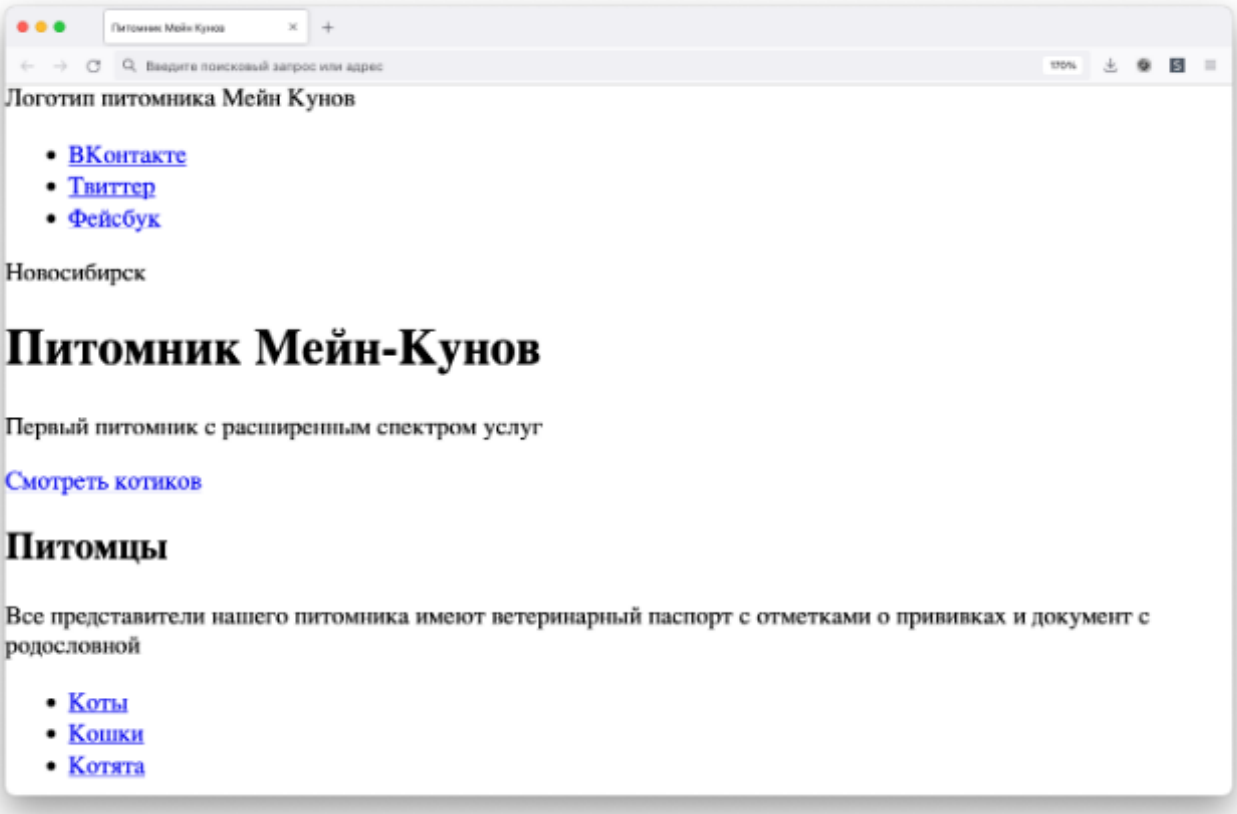
```
index.html — htmlacademy.ru
index.html ×
```

```html
<!DOCTYPE html>
<html lang="ru">

<head>
  <meta charset="UTF-8">
  <meta name="description" content="Питомник Мейн-Кунов в Новосибирске. Первый питомник с расширенным спек
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Питомник Мейн Кунов</title>
  <link rel="stylesheet" href="css/normalize.css">
  <link rel="stylesheet" href="css/style.css">
</head>

<body>
  <header class="header">
    <div class="header__container container">
      <div class="header__wrapper">
        <a class="logo header__logo" aria-label="Логотип питомника Мейн Кунов">
          Логотип питомника Мейн Кунов
        </a>
        <ul class="social header__social">
          <li class="social__item">
            <a href="#" class="social__link">
              ВКонтакте
            </a>
```

А так проект выглядит в браузере:



Базовая стилизация

Начинается самое интересное — работа с внешним видом. Прописываем в `style.css` базовые стили. Для крупных блоков пока ничего не делаем.

Этапы базовой стилизации

— Добавление классов в разметку;

— Подключение нестандартных шрифтов (локально или из сервиса);

— Подключение normalize. css (по желанию);

— Указываем параметры шрифта — название, размер, цвет, жирность;

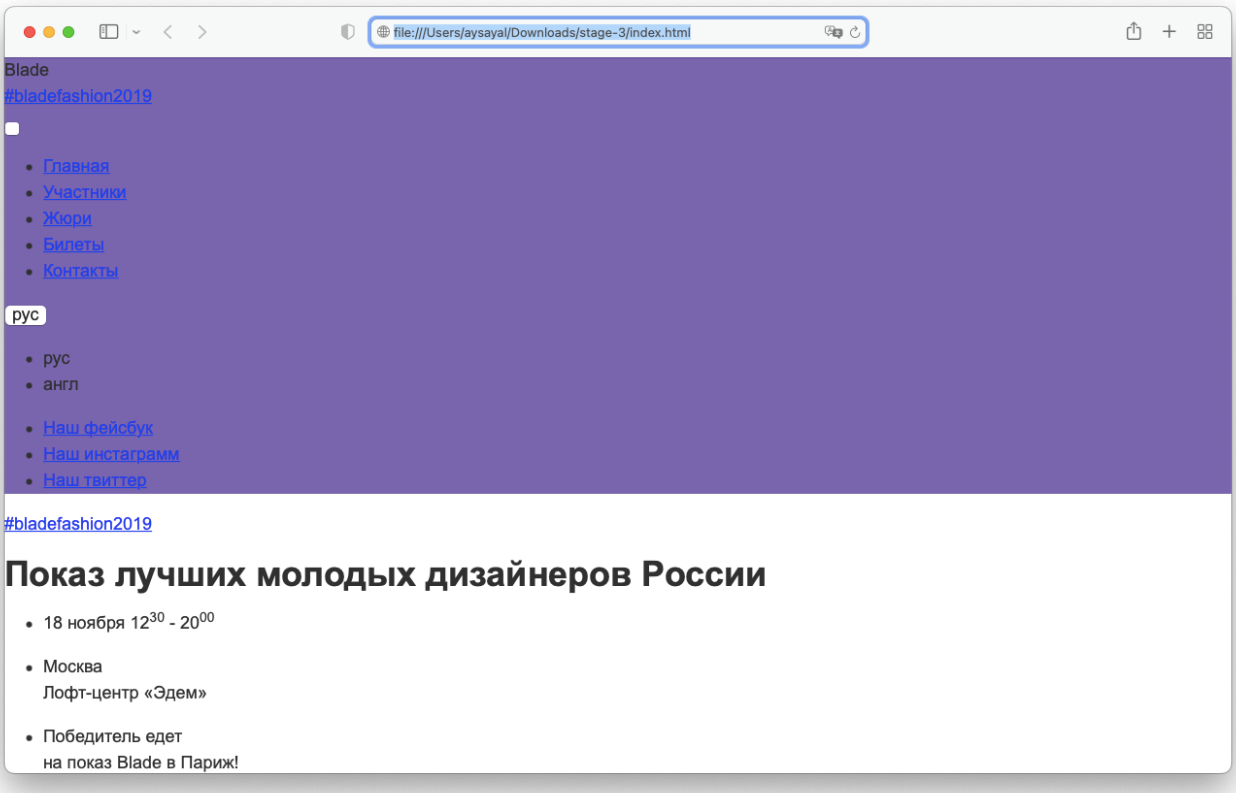
— Указываем высоту строки;

— Описание фоновых параметров (фоновый цвет);

— Описание состояний интерактивных элементов, которые описаны в стайлгайде. На этом этапе задавайте только текстовые параметры и параметры фона;

— Все цвета вынесены в кастомные свойства в селектор `:root.`

Пример проекта с базовой стилизацией:



Работа с графикой

Экспортируйте всю графику из макета в Figma и подключите её в разметке.

SVG-изображения рекомендуется собрать в спрайт и подключить первым элементом в body. Спрайт — это файл, который мы сшили из нескольких графических элементов, например, из иконок. Спрайты экономят запросы к серверу — с ними сайт работает и загружается быстрее. В спрайт сшивается и растровая, и векторная графика.

В папке вашего проекта подготовьте графику: подготовленные изображения разместите в подпапке img в папке проекта, чтобы у вас получилась примерно такая структура проекта:

```
project
| - css
| - fonts
| - img
|      [ваши картинки]
|   favicon.ico
|   index.html
|   catalog.html
```

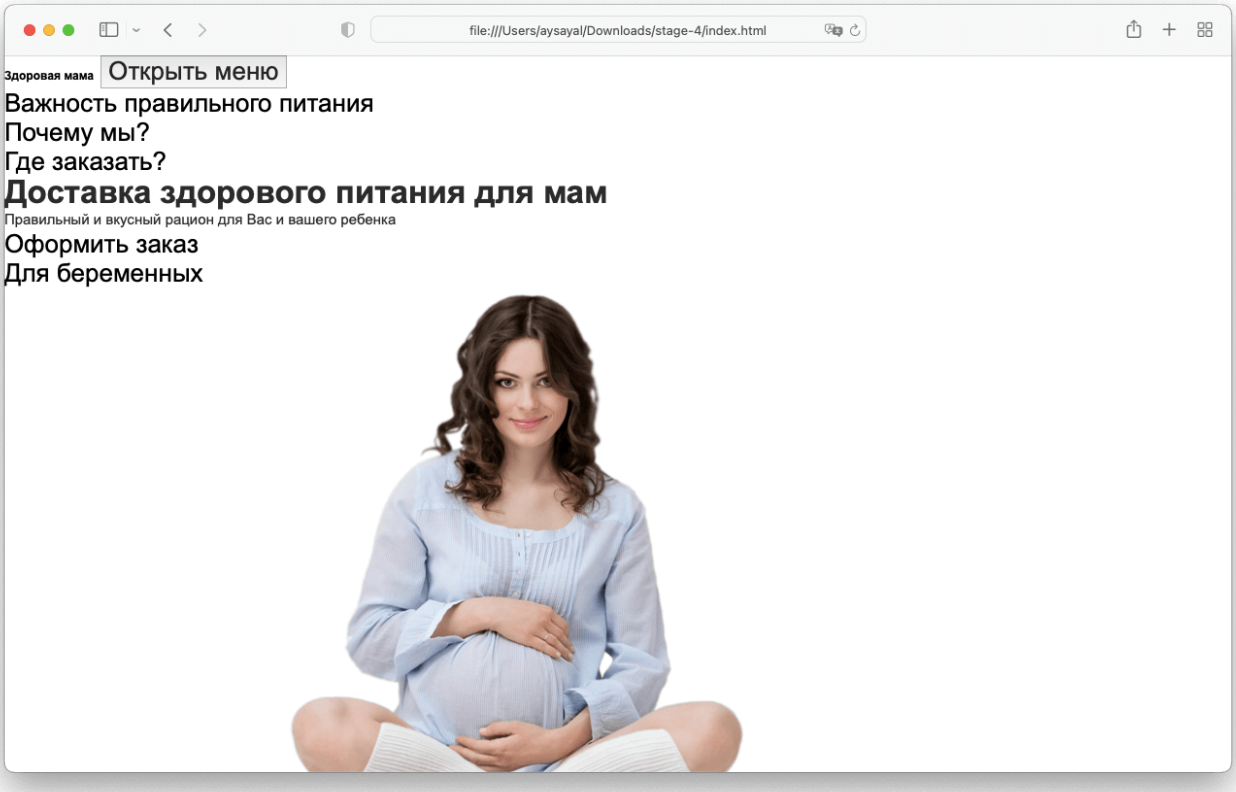
Подключите контентные изображения в разметке. Фоновые и декоративные изображения пока подключать не нужно.

При подключении изображений используйте относительные адреса, обязательно укажите размер картинки без пикселей, а также alt. Например, так:

```
<img src="img/logo.png" width="100" height="50" alt="Девайс">
```

Подключите к проекту фавиконки. favicon. ico размером 32×32 положите в корень проекта и подключите в `<head>`, остальные версии фавиконок делать не обязательно.

Пример проекта в котором подключена графика:



Построение сетки

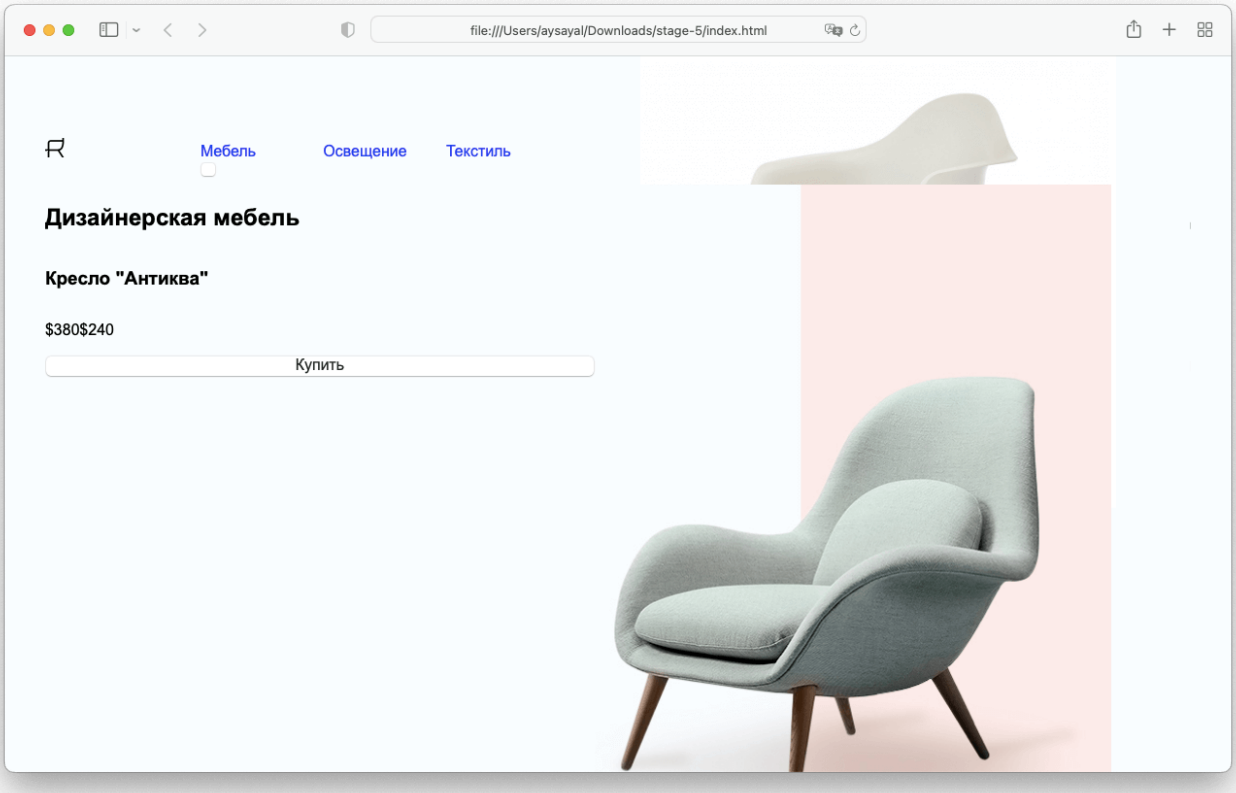
Построение сеток производится в общем стилевом файле style. css.

При работе с созданием крупных структурных сеток, в вёрстке используется подход desktop-first, то есть сайт прежде всего должен корректно отображаться на больших разрешениях экрана.

Расположите элементы страницы по сетке в соответствии с макетом. Для удобства используйте один из следующих способов визуального выделения элементов:

1. С помощью свойства `background-color` с разными цветами для разных блоков;
2. С помощью свойства `outline` (для удобства также можно использовать разные цвета).
3. Свойство `border` лучше не использовать, так как оно влияет на ширину блока и может что-нибудь сломать.

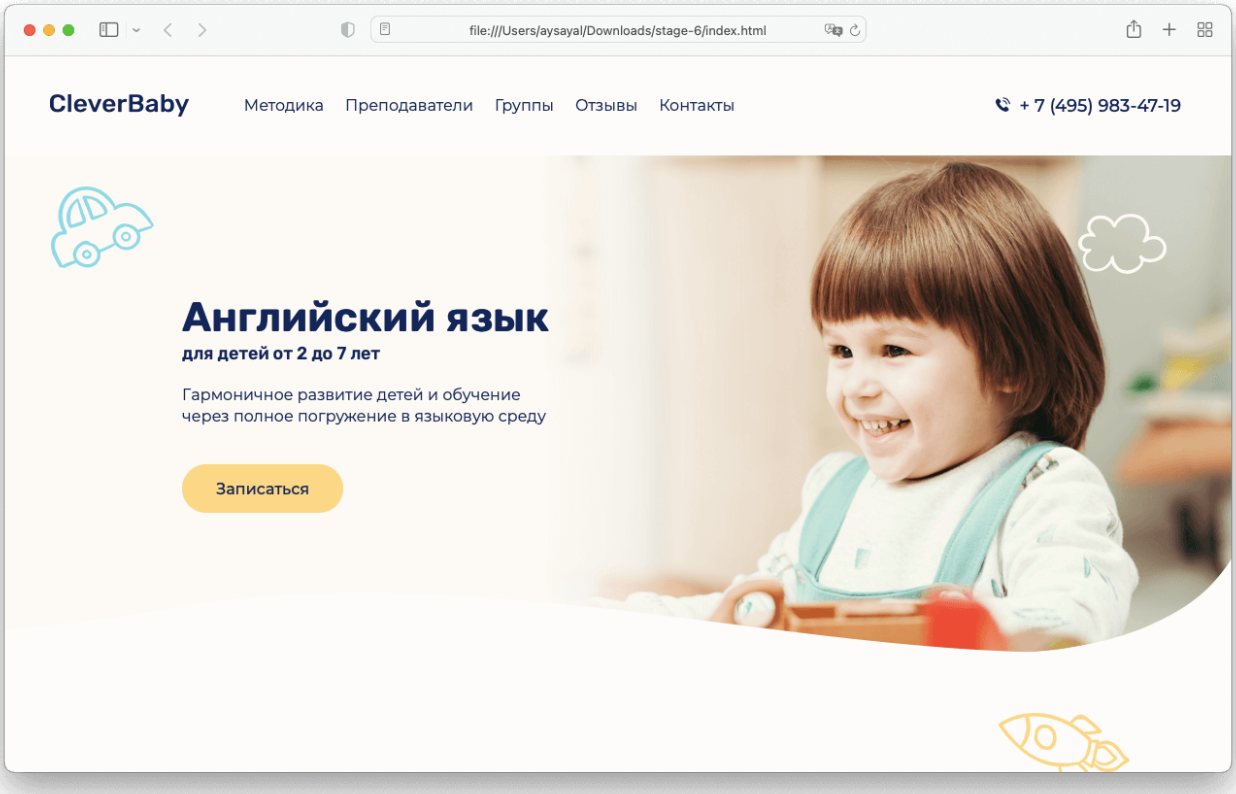
Пример проекта, где уже добавлены сетки:



Добавление декоративных элементов

На этом этапе добавляем мелкие сетки — например, для карточек товара. Подключаем кастомные шрифты и фоновые изображения. В конце оформляем остальные декоративные элементы, которые ещё не стилизовали раньше.

Пример готового проекта, где добавлены декоративные элементы:



Всё почти готово, осталась пара шагов.

Адаптивные сетки и декоративные элементы

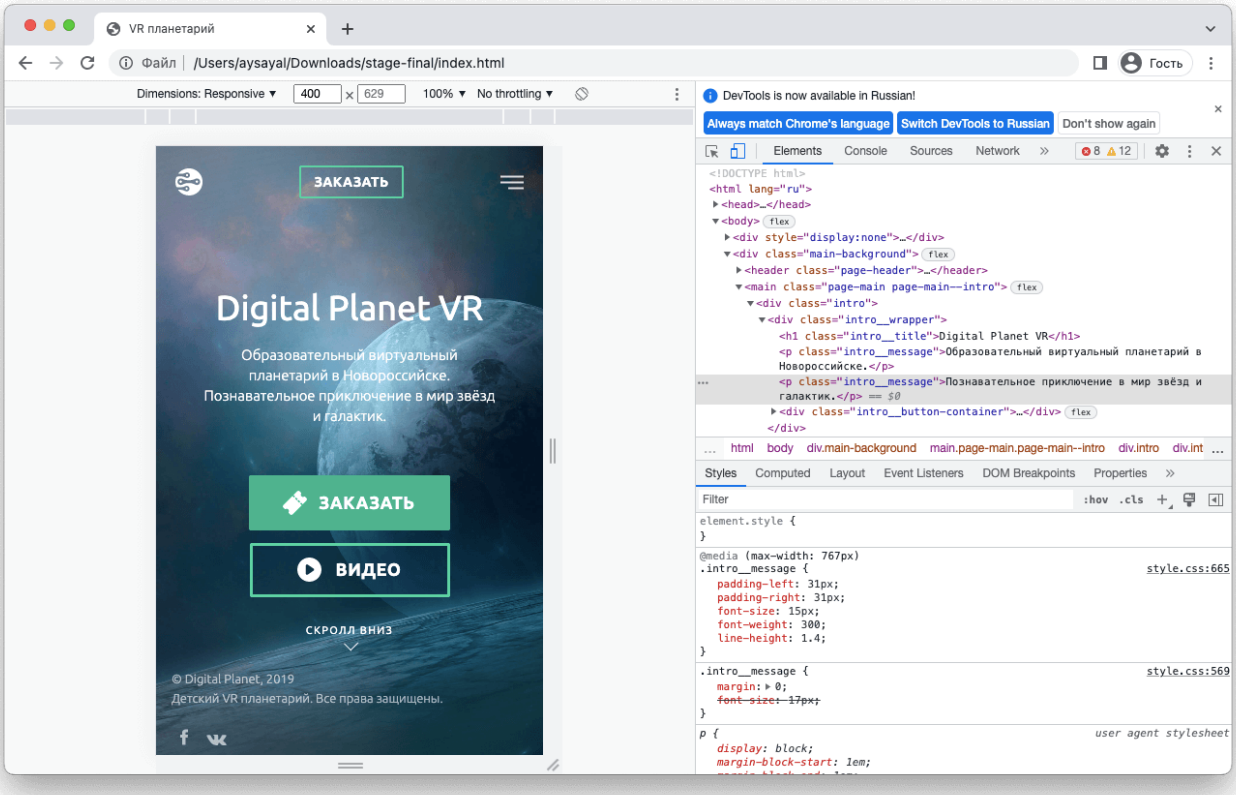
Чтобы сделать хороший адаптивный сайт, нужно понимать много нюансов: как работают вьюпорт, медиавыражения, адаптивные картинки. Эту тему стоит разобрать отдельно, но если хотите, познакомьтесь с ней в блоге Академии.

Адаптивная графика

Ура! Мы и дошли до последнего пункта работы над проектом. Что нам осталось сделать?

1. Подключаем в разметке и стилях адаптивные изображения для разных девайсов и экранов с разной плотностью пикселей;
2. В HTML используем элемент `picture`, с помощью которого подключаем картинки для разных разрешений экрана, для экранов с разной плотностью пикселей, а также webp-варианты картинок для поддерживающих браузеров;
3. В CSS подключаем картинки для экранов с двухкратной плотностью пикселей и для разных разрешений с помощью медиа-выражений.

Пример готового проекта, где настроена адаптивная графика:



Вот мы и подошли к концу. Перед глазами готовый проект, ощущения радостные. Можно смело приступать к верстке нового макета. Ведь, чтобы научиться писать код — нужно писать код. Меньше сомнений — больше практики и всё получится.

Следующие шаги:

- Шаблон простого сайта на HTML
- Шаблон HTML-формы
- Основы дизайна для верстальщиков

Практикум

Тренажёры

Для команд и компаний

Учебник по PHP

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфейсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

Node.js. Профессиональная разработка REST API

Node.js и Nest.js. Микросервисная архитектура

TypeScript. Теория типов

Алгоритмы и структуры данных

Паттерны проектирования

Webpack

Vue.js 3. Разработка клиентских приложений

Git и GitHub

Анимация для фронтендеров

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Услуги

Работа наставником

Для учителей

Стать автором

Информация

Об Академии

О центре карьеры

Остальное

Написать нам

Мероприятия

Форум

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696

Участник

© ООО «Интерактивные обучающие технологии», 2013–2023